❖ Asp.net web server control :-

- The html server controls have properties & method which makes it very easy to migrate from classic asp to asp.net pages employing the html server controls.
- By simply adding runat=”server” attribute we can convert any html control to server control.
- In contrast to html server controls, the asp.net web server controls have an abstracted model which does not map to html elements. Controls attributes differs from html element attributes.
- In html controls, we have a single control (<select> tag) for both list and dropdown which are provided as separate controls in asp.net (listbox & dropdown list)
- The standard controls can be classified in 3 categories :-
  1. Table control
  2. Form control
  3. General control
- Common properties :-
  1. BackColor
  2. BorderColor
  3. BorderStyle
  4. CssClass
  5. Enabled
  6. EnableTheming
  7. ForeColor
  8. Height
  9. Width
  10. SkinID
  11. ID
  12. runat
  13. tooltip
  14. Style

❖ Asp Table controls (<asp: Table>) :-

- Asp contains web server control for the creation of a table, table row and table cell.
- This control is mapped to <table>, <tr> & <td> of html.
- This control is often built programmatically with dynamic contents.
- Each table control is made up of rows which are represented by <asp:TableRow> and is stored in Rows collection of the control.
- Each row is made up of cells which is represented by <asp:TableCell> and is stored in cells collection of the TableRow class.
- E.g :

```
<asp:Table width="500px">
    <asp:TableRow>
        <asp:TableCell> …… </asp:TableCell>
        .
        .
        .
        <asp:TableCell> …… </asp:TableCell>
    </asp:TableRow>
</asp:Table>
```

- Properties of table control :-
  1. BackImageUrl : Specifies URL of background image.
  2. Caption : It is used to give a caption to the table.
  3. CaptionAlign : Specifies the alignment of caption.
  4. CellPadding
  5. CellSpacing
  6. HorizontalAlign
  7. Rows : It specifies the row collection of table.
- Properties of rows :-
  1. Cells : Specifies the cells collection of particular row.
- Properties of cell :-

  1. ColumnSpan

  2. RowSpan

  3. Text : Specifies the text content of the cells.

  4. Wrap : Content of cells will be wrapped by the same value.

- Dynamically creating a table :-

```
Dim T1 As New Table
Dim Tr As New TableRow
Dim Tc As New TableCell
    T1.Rows.Add (Tr)
    Tr.Rows.Add (Tc)
```

❖ Form Controls :-

1. Label Control :-

- It enables us to display a static text on the webpage.
- We can use the text property to display text and it can also be specified programmatically at run time.
- Users can not edit the text of label control at run time.
- We use labels to display validation messages, answer etc.
- We can also make a label as a child control of other controls.
- Syntax:-
  <asp:Label ID=”lbl1” runat=”server” Text=””> </asp:Label>
- E.g Content.aspx :-
  Enter your name:
  <asp:Textbox ID=”txtname” runat=”server”>
  </asp:Textbox>
  <asp:Label ID=”lbl1” runat=”server” Text=””> </asp:Label>

  <asp:Button ID=”btnsubmit” runat=”server” Text=”Submit”> </asp:Button>
- Content.aspx.vb :-

```
Partial Class Content
        Inherits System.Web.UI.Page
    Protected Sub btnsubmit_Click (ByVal Sender As object, ByVal e As System.EventArgs) Handles btnsubmit.Click
        Dim str As String
        Str=”My Name is : “ & txtname.text
        lbl.Text=str
    End Sub
End Class
```

- Properties of label control :-
  1. ID
  2. TabIndex
  3. Text
  4. ToolTip
  5. Visible
  6. AccessKey
  7. Runat

2. Literal Control :-

- When we want to render text and control directly on a page without any markup, we use a literal control.
- It reserves a location on the web page to display static text.
- We can edit the text in a literal programmatically at run time.
- It is similar to label exception that is does not support any formatting properties.
- Properties of literal control :-
  1. ID
  2. Text
  3. Visible
  4. EnableViewState
  5. Runat
  6. Mode :- It specifies how the content in the literal will be rendered on the page.
     - Mode=”PassThrough” displays the content of control without encoding. E.g <h1>Hello</h1> o/p : Hello
     - Mode=”Encode” displays the content of literal after encoding. E.g Hello o/p : **Hello**
     - Mode=”Transforms” it displays the control after removing markup that is not supported by requesting client.
- E.g <asp:Literal runat=”server” ID=”literal1”></asp:Literal>

❖ Difference between Label and Literal :-

Label :

1. Default text property in label is set to label 1.
2. Label tag is rendered as span tag in html.
3. We can apply all formatting styles on a label.
4. We can not use Asp label in between title tag.

Literal :

1. Default text property is set to null.
2. Literal tag does not rendered as html element.
3. We can not apply any formatting.
4. We can use Literal between title tag.

3. Textbox Control :-

- It is like the html textbox control which allows the user to enter some text.
- The Asp.net textbox control is flexible and can be configured to support single line, multiple or password modes.
- Properties of textbox control :-

  1. AutoPostBack :

     It is a Boolean property which specifies whether the control will post the contents automatically to the server when the content of control changes.

  2. Columns :

     It specifies the width of textbox.

  3. Rows :

     It specifies the number of rows when the text mode is multiline.

  4. TextMode = “SingleLine/MultiLine/Password”

  5. ReadyOnly :

     It is a Boolean value which specifies whether the user can edit the content of textbox.

  6. Wrap :

     It is a Boolean value which specifies whether to wrap text or not.

  7. MaxLength :

     It specifies the maximum number of character that can be entered in the textbox.

  8. Text

  9. Visible

4. Link Button Control (<asp:LinkButton>) : -

- <asp:LinkButton> control is similar to hyperlink of the html control but is same as the button control in terms of functionality.
- We can write code on the click of a link button.
- Properties of Link Button control :-

  1. Text: Specifies the text to display within the linkbutton.

  2. PostBackUrl: It specifies the URL of the page to post form the current page.

  3. CausesValidation: By default a page is validate, when a button control is clicked. To prevent a page or a control form being validated when clicking on a button, set this property to false.

- E.g

  <asp:LinkButton runat=”server” ID=”Link1”
  PostBackUrl=”~/Login.aspx”></asp:LinkButton>
  Partial Class LinkB
  Inherits System.Web.UI.Page
  Protected Sub Link_Click (ByVal sender As object, ByVal e As
  System.EventArgs) Handles Link1.Click
  Response.Redirect (“Home.Aspx”)
  End Sub
  End Class

5. Button Control :

- It allows us to create a push button on a web form.
- Bydefault, buttons submit the page to the server and there it is processed along with some events.
- Two types of buttons can be created.
  1. Submit Button
  2. Command Button
- A submit button submits the page to the server by executing the instructions attached to the buttons event handler.
- A command button has a command name specified by CommnadName property.
- This allows us to create multiple buttons on a webpage and programmatically determine which button is clicked by handling the command event.
- Properties of Button Control :-
  1. CausesValidation
  2. Text
  3. PostBackUrl
  4. CommandName : It specifies the command associated with command event.
  5. OnClientClick : The name of the function to be executed when the button is clicked.
- Events of Button Control :-
  1. Click
  2. Command
- E.g Content.aspx

  <form runat=”server”>
  A<asp:Textbox runat=”server” ID=”txtA”>

```
    </asp:Textbox>
  B<asp:Textbox runat="server" ID="txtB">
    </asp:Textbox>
  <asp:Button ID="btnA" CommandName="A" runat="server" Text="+"/>
  <asp:Button ID="btnS" CommandName="S" runat="server" Text="-"/>
  <asp:Button ID="btnM" CommandName="M" runat="server" Text="*"/>
  <asp:Button ID="btnD" CommandName="D" runat="server" Text="/"/>
  ANS<asp:Textbox runat="server" ID="txtAns">
    </asp:Textbox>
</form>
```

- Content.aspx.vb:

```
Partial Class Content
        Inherits System.Web.UI.Page
    Protected Sub btnA_Command (ByVal sender As object, ByVal e As System.Web.UI.webcontrols.CommandEventArgs) Handles btnA.Command, BtnS.Comamnd, btnM.Comamnd, btnD.Command
        Dim no1, no2 As Double
            no1=txtA.Text
            no2=txtB.Text
        If e.CommandName="A" Then
            txtAns.Text = no1 + no2
        Else If e.CommandName="S" Then
            txtAns.Text = no1 - no2
        Else If e.CommandName="M" Then
            txtAns.Text = no1 * no2
        Else
            txtAns.Text = no1 / no2
        End If
    End Sub
End Class
```

6. ImageButton Control :-

- It is used to display clickable image. To set the image for this control, we use the ImageUrl property.
- It also supports both click and command events like button control.

- Properties of ImageButton control :-

  1. PostBackUrl : It specifies URL of the page where the current page's content will be posted to.

  2. ImageUrl : It Specifies the URL of the image to be set on the button.

  3. CommandName : The name of the command associated with the command event.

7. Radio Button Control :-

- It is used to select a single option from a list of given items. It is also known as option button.
- We use radio buttons for attributes like gender, qualification, category, mcqs etc where multiple options are given but we select only one.
- For this, we create a radio button group such that whenever one button is selected others get unselected.
- Properties of Radio Button Control :-

  1. Checked: A Boolean value which specifies whether the radio button is checked or not.

  2. GroupName: The name of the group to which a radio button belongs. So that only one can be selected at a time.

  3. AutoPostBack: A Boolean value which specifies whether the control will postback immediately after the checked property is changed.

  4. Text: Specifies the text next to the radio button.

  5. TextAlign: Specifies the alignment of the button.
- Events of Radio Button control :-

  1. CheckedChanged
- E.g RadioBtnEg.aspx :

  <asp:RadioButton Text="FYBCA" GroupName="selyr" runat="server" ID="rdFy" AutoPostBack="True"></asp:RadioButton>

  <asp:RadioButton Text="SYBCA" GroupName="selyr" runat="server" ID="rdSy" AutoPostBack="True"></asp:RadioButton>

  <asp:RadioButton Text="TYBCA" GroupName="selyr" runat="server" ID="rdTy" AutoPostBack="True"></asp:RadioButton>

  <asp:Image Width="300px" Height="200px" runat="server" ID="imgyr" />
- RadioBtnEg.aspx.vb :

  Partial Class RadioBtnEg

  Inherits System.Web.UI.Page

```
    Protected Sub rdFy_CheckedChanged (ByVal sender AS object, ByVal e As System.EventArgs) Handles rdFy.CheckedChanged
        If rdFy.Checked = True then
            Imgyr.ImageUrl = "~/images/Fy.jpg"
        End If
    End Sub
End Class
```

8. CheckBox Control :-

- It is a web server controls that provides user to switch between yes or no or true or false options.
- It is ideally used to give multiple options to the user.
- Properties of Checkbox control :-
    1. AutoPostBack
    2. Checked
    3. Text
- Events :-
    1. CheckedChanged
- E.g CheckBoxEg.aspx :

```
<asp:CheckBox ID="chkFy" runat="server" Text="FYBCA" AutoPostBack="True" />
<asp:CheckBox ID="chkSy" runat="server" Text="SYBCA" AutoPostBack="True" />
<asp:CheckBox ID="chkTy" runat="server" Text="TYBCA" AutoPostBack="True" />
<asp:Panel ID="panFy" Visible="false" runat="server" GroupingText="Fy Subjects">
    FYBCA Subjects
</asp:Panel>
<asp:Panel ID="panSy" Visible="false" runat="server" GroupingText="Sy Subjects">
    SYBCA Subjects
</asp:Panel>
<asp:Panel ID="panTy" Visible="false" runat="server" GroupingText="Ty Subjects">
```

TYBCA Subjects

</asp:Panel>

- CheckBoxEg.aspx.vb :

```
Partial Class CheckBoxEg
        Inherits System.Web.UI.Page
    Protected Sub chkFy_CheckedChanged (ByVal sender As object, ByVal e As
System.EventArgs) Handles chkFy.CheckedChanged
            panFy.Visible = True
    End Sub
End Class
```

9. File Upload Control :-

- It allows the users to upload files and send it to the server.
- It is useful for uploading features, text file etc.
- A user can select a file by clicking on the browser button and locating the file from choose file dailogbox.
- The file upload control does not save a file automatically to the server. A developer needs to upload it explicitly by writing a code to submit a file.
- We have a method Save As which saves the contents of a file to a specified path on the server.
- Before calling this method we use the 'HasFile' property to check whether the file upload control contains a file or not.
- Properties of File upload control :-
  1. FileBytes : Returns the no. of bytes of the file as an array.
  2. FileContent : Returns the content of file as a stream.
  3. FileName : Returns the name of the file uploaded.
  4. HasFile : Specifies whether a file is selected or not.
- E.g Content.aspx :-

```
<asp:FileUpload ID="f1" runat="server" />
<asp:Button ID="Fupload" runat="server" Text="Upload" />
<asp:Label ID="lblFile" runat="server"></asp:Label>
```

- Content.aspx.vb :-

```
Partial Class FileEg
        Inherits System.Web.UI.Page
```

```
        Protected Sub Fupload_Click (ByVal sender As object, ByVal e As
System.EventArgs) Handles Fupload.Click
                Str= server.MapPath (“images”)
                If f1.HashFile = true then
                        Str = Str & “\” & f1.FileName
                        f1.SaveAs (Str)
                        lblFile.Text = “File upload successfully “
                Else
                        lblFile.Text = “File does not exist “
                End If
        End Sub
End Class
```

- The Server.MapPath method traces the location of given path (images) where we want to upload the given file (f1).
- Then after the SaveAs method saves the file (uploads) on the mapped path on the server.

❖ Hyperlink Control :-

- A hyperlink control is used to create a hyperlink in ASP.NET.
- We can specify the hyperlink text using the text property.
- We can also display an image instead of hyperlink by using ImageUrl property.
- Properties of Hyperlink control :-
  1. ImageUrl: Specifies the URL of the image to be displayed as a hyperlink.
  2. NavigateUrl: Specifies the URL of the page to be navigated to.
  3. Target: It specifies the target frame of the URL. (_blank, _parent, _top, _self)
- Syntax :-

```
<asp:Hyperlink ID=”hypLnk1” runat=”server” ImageUrl=”images\xyz.jpg”
NavigateUrl=”Home.aspx”>
</asp:Hyperlink>
```